Defense Technical Information Center
Compilation Part Notice

ADP010846

TITLE: Basic Distributed Control Model and Technology for Mobile Crisis Reaction Forces and their United Air Defense

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: Systems Concepts for Integrated Air Defense of Multinational Mobile Crisis Reaction Forces [Concepts de systemes pour la defense aerienne integree de forces internationales mobiles d'intervention en situation de crise]

To order the complete compilation report, use: ADA391354

The following component part numbers comprise the compilation report:

ADP010843 thru ADP010864

UNCLASSIFIED

Basic Distributed Control Model and Technology for Mobile Crisis Reaction Forces and their United Air Defense

Peter S. Sapaty
Institute of Mathematical Machines and Systems
National Academy of Sciences of Ukraine
Glushkova Avenue 42, 252187 Kiev, Ukraine
+380-44-2665023, +380-44-2666457 (fax)
sapaty@immsp.kiev.ua, psapaty@hotmail.com

Abstract: The paper investigates the use of the distributed processing and control model and technology, WAVE, operating in open computer networks and providing integral solutions of complex problems on a high semantic level, for a variety of system organization and management levels and tasks in relation to the mobile Crisis Reaction Forces and their integrated air defense. The technology hides most of traditional communication and organization routines like message passing, intelligent agents, mobile agents, remote procedure calls, remote method invocation, distributed object brokers, etc., within the system implementation, allowing application programs to be extremely powerful and compact. Based on a free migration of cooperative program code in both physical and virtual worlds and parallel spatial matching of the systems navigated, while creating and modifying the systems themselves, the technology allows for an unlimited scaling, and works equally well with any number of computers and any network topologies, which may be loose, dynamic, and open.

1 Introduction

Crises appear in different, often unpredictable points of the world. Their scale may range from local regions in particular countries to the whole continents or even world-wide. Typical examples may include forest fires, earthquakes, flooding, nuclear plant accidents, viral diseases, ethnic conflicts, coups, etc. Separate countries with often limited national natural, technological, technical, and human resources may be unable to withstand the problems occurred. That is why critical may be the use of united, rapidly composable and deployable international forces, where the dissimilar resources from different countries should be united within the common disaster or conflict relief mission.

Efficient computerization of the international Crisis Reaction Forces (CRF), which may be required to operate in highly dynamic and hostile environments, is vital for keeping their integrity, external and internal controllability, and high survivability. Radically new information and networking technologies may be needed oriented on solving the mission problems in a parallel and distributed mode, pursuing both local and global, often dynamically changeable, goals. These technologies should allow for an efficient merge of distributed heterogeneous information and command and control systems of constituent forces from different nations, supply and re-supply of limited, both crisis relief and own mission support resources, runtime re-composition and recovery from indiscriminate damages, high overall awareness with collective decision-making, and quick reaction on multiple external threats.

Advanced military-oriented mobile CRF, due to high level system organization based on computerization and networking, may be capable of withstanding considerable enemy forces and solving very complex military tasks in a highly flexible mode, often without involvement of traditional heavy armor techniques. As an example of a possible development of CRF may be the organization of Future Combat Systems (FCS) – a US army vision of 2025, with the related project just announced by DARPA. FCS will represent strategically deployable, tactically superior and sustainable force, with quick reaction capability, air-mobile operation, lightweight units (not more than 20 tons each), increased lethality, survivability, mobility, and deployability. They are also expected to have effective distribution of sensors and integration of robots into the force, overall networked organization and networked fire, high common situational awareness and understanding.

Paper presented at the RTO SCI Symposium on "System Concepts for Integrated Air Defense of Multinational Mobile Crisis Reaction Forces", held in Valencia, Spain, 22-24 May 2000, and published in RTO MP-063.

A very important but extremely complex problem for international mobile CRF, with their distributed, changeable and dynamic organization and structure, is an efficient air defense in order to protect from aerial attacks, rockets, artillery, mortars, and aerial observation, always preserving integrity and uninterrupted functionality. The following tasks are among the many to be solved efficiently for the air defense of CRF: quick discovery of hostile aerial objects throughout the internationally controlled region; identification, tracking and behavioral analysis of multiple targets; global assessment of the aerial threat with making collective decisions; optimization of the use of distributed antimissile weapons; interface with other weapon systems and manned or unmanned fighter planes; participation in the higher-level battlefield operations and management.

The rest of the paper is organized as follows.

Section 2 shows the need of radically new information and networking technologies capable of supporting such dynamic distributed systems as CRF, because traditional cultures and approaches to organization of distributed networking projects inevitably lead to huge communication, synchronization and control overheads, numerous seams and multilingual patches. The interpreted WAVE distributed control model, language and technology, allowing for high-level semantic solutions in the space navigation mode, with unlimited program code mobility, may be a real candidate for the integration of CRF-like systems. It hides most of traditional organization and management routines within the language implementation, making parallel application programs extremely simple, powerful, and compact.

Section 3 gives a brief overview of the extended WAVE language capable of describing parallel and distributed solutions in both physical and virtual worlds. General organization of the recursive language syntax and basics of semantics are given, with details concerning representation of space and movement in it, data structures, different types of spatial variables which may be stationary or mobile, elementary operations, and control rules setting proper constraints coordinating parallel conquest of space.

Section 4 provides some information about the implementation of WAVE by a network of the language interpreters, and describes a general organization of the interpreter which can execute parts of WAVE programs (waves) while sending other parts to another interpreters. Forward and backward data and control echoes being other communication messages, with the overall integrity of self-evolving spatial processes provided by the dynamic distributed track system. The interpreters can also reside on mobile platforms, being invoked at runtime in proper locations on the demand of space-navigating waves.

In **Section 5**, a review of a number of existing practical applications of the WAVE technology is provided, which include integration of distributed databases, intelligent network management, distributed interactive simulation of dynamic systems like battlefields, distributed multiuser virtual reality, road traffic management, and modeling collective behavior of robots. A number of projects have been successfully demonstrated via the Internet with computers distributed between different countries.

Section 6 describes advantages of organization of mobile CRF in WAVE, among which high integrity, flexibility, and external controllability may be of particular importance for advanced military campaigns, with WAVE interpreter being installed in both manned and unmanned platforms. The section provides solutions in WAVE of some basic CRF management operations. First, it describes parallel creation and reconfiguration of a hierarchical command and control infrastructure establishing subordination between the army units. Second, it defines and recursively implements a typical command and control process, with commands being executed at different levels, and modified and sent further down to the subordinate levels, with the execution confirmation ascending the hierarchy. Third, it provides an exemplary solution of a typical resource management and distribution task, where some limited resource from a central storage is physically delivered to army units that requested it, where the decision concerning a particular amount allowed to each requesting unit is made via the established command hierarchy. The section concludes with a multiple management scenario, where different local scenarios, on behalf of army units, regularly interact with each other via the infrastructure, in order to find a balanced distributed solution satisfactory to all parties, one scenario performing a global moderation.

Section 7 is concerned with the use of WAVE for organization of a united air defense of CRF. The main task here is to integrate into one system the radar stations belonging to different army units, which may be of short range and not capable of covering the entire air space alone. To keep the whole space under control, the stations must communicate. To optimize communications, a radar neighborhood infrastructure is dynamically created, maintained, and regularly updated in WAVE, through which most communications between radar stations should take place. The main mobile tracking algorithm in WAVE is demonstrated which, after having seized an aerial object, follows it via the neighborhood infrastructure, providing the object's handover between radar stations. Many such objects may be tracked independently and in parallel. Possible payload of this basic algorithm is discussed like collection of the object's itinerary, analysis of its behavior, and invocation of antimissile weapons. The latter may also need networked solutions to be towed by mobile software agents to the proper regions in space to meet the targets.

In case of being lost, the objects can be rediscovered by other stations, with new mobile processes uniquely assigned to them. Using additional stationary and/or mobile coordination processes in WAVE, interacting with the mobile tracking processes, it is possible to make a non-local optimization of the use of limited antimissile hardware, keep an overall awareness of the level of aerial threats, as well as find suitable system solutions interactively. Many other air defense related problems of mobile CRF may be described and solved in WAVE in a similar way, while keeping the whole system as a highly intelligent reactive and self-protective distributed brain.

Section 8 concludes the paper, summarizing main features of WAVE and outlining prospects of its use for advanced military systems.

2 In a Search for the New Organization and Coordination Technologies: WAVE

CRF will need distributed system solutions, as different, often dissimilar, pieces of the mission information may potentially be located in any computer and in any vehicle, and no central databases or centralized processing, control and management may be welcome, in order to reduce the system's vulnerability to a minimum, as any parts of the system may be indiscriminately damaged in a campaign.

2.1 Traditional overhead of networked solutions

In single-processor solutions, written in traditional application languages, everything is used to be at hand, and full and direct control over any resources is guaranteed. Changing any strategy or tactics may need a single operation only (just changing the program counter on a machine level). For CRF, to provide fully distributed solutions, the whole project should be considered as broken into many pieces (see Fig. 1) which have to be distributed in space and, moreover, may move physically, constantly changing absolute and relative to each other positions.

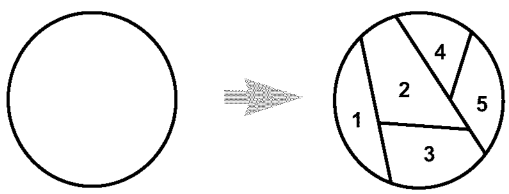

Fig. 1. Breaking an integral single machine solution into multiple pieces for distribution

Making these pieces, located now in different computers, work together properly, often results in a huge communication, synchronization and (multilevel) control overhead, with involvement of other languages and techniques for gluing and linking. This inevitably leads to system heterogeneity, multiple patches, and seams. The overhead often outweighs, by orders of magnitude, the useful work done in a single machine solution (see Fig. 2). Distributed networked systems also take considerably longer time to understand, design, debug, test, and produce.

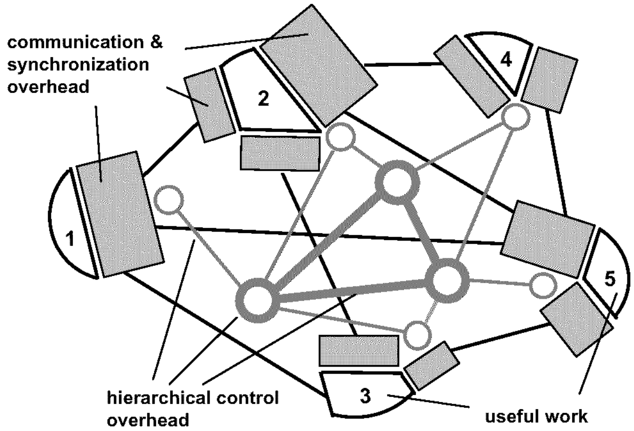

Fig. 2. Traditional communication and control overhead of distributed networked solutions

All this puts forward, especially for such dynamic distributed systems as CRF, a necessity of special formalisms and technologies that would allow to reduce the difficulty of achieving distributed computerized solutions, while making them comparable in complexity, say, to programming on a single machine. Such formalisms should orient primarily on integration, control and management, rather than computation, and should allow for the description of distributed system solutions on much higher than traditional levels, in order to hide the diversity of communication routines, patches and seams, currently needed to be programmed explicitly in traditional distributed projects, within the implementation. The WAVE model and technology discussed in this paper are just oriented on meeting and fulfilling these objectives.

2. 2 Distributed computation and control in WAVE

WAVE is a special parallel and distributed coordination and computation model operating in open networks [1]. It is technically based on a universal control and data processing module, communicating copies of which are distributed throughout the system to be controlled. The static or dynamic network of the modules is governed at the top by a high level distributed processing language, WAVE, allowing for parallel navigation and supervision of the whole system or its arbitrary parts and interaction with multiple users. The said modules being copies of the WAVE language interpreter, which may have both software and hardware implementation. Navigating in space, WAVE also creates persistent distributed virtual, or knowledge, networks, shared by differed users and other navigational processes, effectively supporting scaleable control, simulation, and virtual reality systems. Any other systems and technologies, in a variety of other languages, can be accessed, integrated, and controlled in WAVE.

The WAVE language describes a stepwise parallel flooding, or coverage, of physical or virtual spaces, as depicted symbolically in Fig. 3, providing distributed seamless solutions of complex system problems without traditional message passing, RPC, clients-servers, agents, mobile agents, objects, etc., usually causing huge programming overhead. These and other techniques are used *on implementation levels only*, completely relieving programmer from traditional routines and making coordination and management programs extremely powerful and compact.

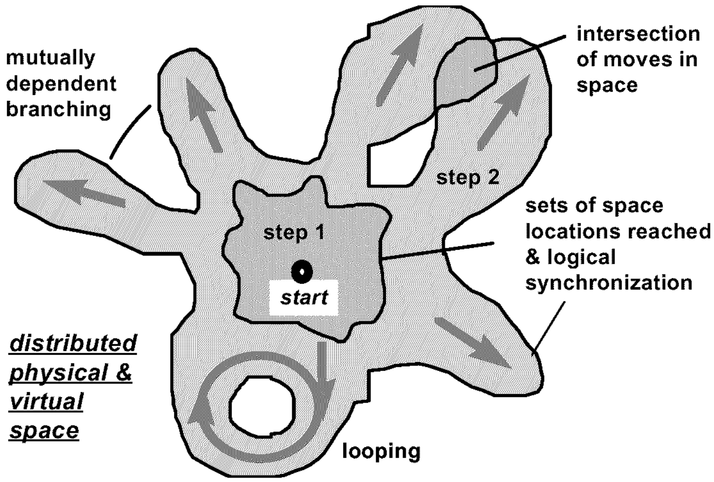

Fig. 3. Seamless integral distributed navigation & supervision & processing in WAVE

3 Extended WAVE Language

3.1 General organization

WAVE language programs (called "waves") describe multiple actions in distributed, both physical and virtual worlds (respectively, PW and VW), as well as in their combinations. Every elementary activity provided by the language is said to be performed in a *node* which represents a proper point of physical or virtual space. Nodes may be temporary, just identifying locations where and when the activity is applied, or persistent, expressing established concepts or facts which may exist independently, as they are, and remain arbitrary long time.

Waves, starting from some point (node), propagate in space, sequentially or in parallel, causing the appearance of new nodes, or entering the already existing nodes, created (and, possibly, occupied) by other waves. During the propagation, waves may form and leave links between nodes reflecting different kinds of inter-nodal relations. Arbitrary actions can be performed in nodes, including changing the node and/or link contents, as well as removing the nodes together with the adjacent links. Additional, temporary data may be left at nodes and shared with other waves, while other data can move with waves further, as their property. The data may represent pure information, physical matter or combination thereof. Staying and doing computations in nodes, waves may also search physical or virtual spaces to a given depth, by launching subordinate waves for collecting and bringing back (remote) data, to be processed further in the current nodes. They can also change data in (remote) locations and interact with other waves through the shared access to the same nodes.

Multiple waves, evolving simultaneously in the same or in different nodes, may be covered by distributed recursive control set up by *rules*, evolving and propagating in space together with the waves. Rules coordinate cooperative, competitive or independent conquest & supervision of space in both breadth and depth mode. They also provide creation of distributed virtual networks by linking the newly formed or already existing nodes, perform spatial merging and parallel processing of multiple remote results, as well as allow WAVE to be used as a traditional sequential or parallel programming language.

WAVE has an extremely simple, recursive, syntax, shown in Figure 4, where coordinated propagation in space can be integrated with the collection, return, and processing in the space locations of data obtained in another, possibly, remote locations, via another space propagation. Words in Fig. 4 represent syntactic categories, braces show zero or more repetitions of a construct with a delimiter at the right, square brackets identify an optional construct, and a vertical bar separates alternatives. Others being the language symbols: semicolon allows for a sequential, while comma for parallel or arbitrary order invocation of waves (under some rules, comma may serve as just a separator between branches), and parentheses are used for structuring waves.

Sequential steps, or *zones*, develop from all locations/nodes of the set of nodes reached (SNR) by a previous zone, while parallel steps, or *moves*, develop from the same nodes, adding their SNRs to the SNR of the zone. SNR may contain nodes repeatedly, reflecting splitting & intersection of waves in the same locations, and subsequent SNRs may have nodes of the previous SNRs, thus allowing for loops in space.

wave	→	{ zone ; }
zone	→	{ move , }
move	→	value | { move act } | [rule] (wave)
value	→	constant | variable
variable	→	nodal | frontal | environmental
act	→	control_act | fusion_act
rule	→	forward_rule | echo_rule

Fig 4. WAVE language recursive space-navigating syntax

Moves have a recursive definition and can be of three types. First, they can point at a resulting value directly (as a constant or variable). Second, they can form space navigating & data processing expressions consisting of arbitrary moves separated by elementary operations, or acts, where moves may return (local or remote) results on the demand of acts, or assign the results to (local or remote) variables, or do the both. Third, they can themselves be arbitrary waves (in parentheses), optionally prefixed by control rules. This simple recursive definition of moves allows for an extremely powerful and compact expression of arbitrary complex, parallel and distributed space navigation, data processing and control operations, which can be carried out in a fully distributed and highly parallel mode.

3.2 Some language details

Representation of space. Any point in a continuous PW may be represented and reached by its absolute coordinates, as a *node*. Moving to other points/nodes can use the absolute destination coordinates or coordinate shifts from the previous node. It is also possible to move to the already existing nodes, reached by other activities within a certain range from a given center point, and many such nodes may be reentered simultaneously. PW nodes are temporary and exist only if activities (waves) stay in them. VW nodes have names, being also their contents, by which they may be referred to globally. They also have unique addresses, which may be used for their quick direct access. Nodes may be interconnected by links, links having names or contents too. Movement in VW may be done by direct hops using node names and addresses, or by following links (using their names and orientation), this movement can be done in a selective or broadcasting mode. VW nodes are persistent: after creation, they exist until deleted explicitly; this is accompanied by a deletion of adjacent links. Node and link names may be any strings of code, including programs to be executed (in WAVE or any other language). VW nodes may be associated with proper locations in the PW, and PW nodes may have addresses and may be dynamically linked to VW or other PW nodes, resulting in a deep, seamless, integration of both worlds within the same space-conquering & processing formalism.

Vectors. The only datastructure of WAVE, symbolically called a *vector*, is a dynamic collection of elements separated by a comma and enclosed in parentheses, if more than one element. Vectors have dual data & program nature, being treated as evaluated waves. All acts in the language are defined over vectors and operate on their multiple values. The latter may be numbers or strings where strings in single quotes represent information (braces are used to represent program strings to be optimized), and in double quotes -- physical matter. Different acts over vectors treat them either as ordered sequences or sets. Special syntax of waves and the possibility of creating arbitrary virtual networks navigated subsequently by other waves, allow us to work easily with arbitrary datastructures of any existing or imaginary languages, and in a highly parallel and fully distributed mode.

Variables. There are three types of distributed dynamic variables the spreading waves operate with. *Nodal*, or stationary, variables (identifiers prefixed by N or M) are created in nodes and may remain there, shared by different waves traversing the nodes. *Frontal* variables (identifiers starting with F) travel with waves, replicating when waves split. *Environmental* variables (each having a special name) access different properties of the navigated PW and VW worlds, also providing impact on the worlds in their different points.

Acts. Acts, operating on their left and right operands, may form arbitrary complex world navigating & data processing expressions, directly working with both local and remote values. *Control acts* permit, direct, or halt program and data flow through nodes where they are interpreted, can inject new executable wave code into the program. *Fusion acts* provide data processing, returning results to be used by other fusion or control acts, they can also access external systems. A number of fusion acts can be applied to physical objects and their storage, while other operations on physical matter may require special external functions.

Rules. *Rules* establish a variety of constraints upon the distributed development of waves. *Forward rules* coordinate spreading of waves in space. Among them, *branching rules* split the wave and coordinate parallel or sequential development of different branches. Other forward rules include repetition of spatial navigation, remote logical synchronization, protecting access to common resources, granting autonomy to waves, allowing spreading waves to create distributed networks, etc. *Echo rules* accumulate, generalize, and process states or results (including remote) reached or produced by the embraced wave, returning them to the rule activation node for further assessment and processing. Rules operate using a powerful internal track & echo system allowing for a generalized, as well as detailed, supervision of distributed solutions, which may spread over arbitrary large territories.

4 WAVE Implementation

The WAVE language is executed by a network of software or hardware interpreters interacting copies of which should be installed in different parts of the systems to be managed.

4.1 The interpreter architecture

General organization of the interpreter is shown in Figure 5. It consists of the three main functional modules: *parser*, *data processor*, and *control processor*. These divide between themselves the responsibility of handling different interpretation data structures, performing specialized operations of the language parsing, execution, and exchanges with other interpreters, where the operations in different units may overlap in time. The interpreter, if to be installed in (manned or unmanned) mobile platforms, may also contain special software or hardware modules coordinating continuous motion in space, providing vision of the environment, manipulating with physical objects, etc., as well as making communication with other interpreters using radio, radar, laser or sonar channels. The communication module finds other interpreters (vehicles) in space by the given search parameters and exchanges with them waves, echoes and remote results via incoming and outgoing queues. Links between the modules are shown in Figure 5 together with main types of information moving along them.

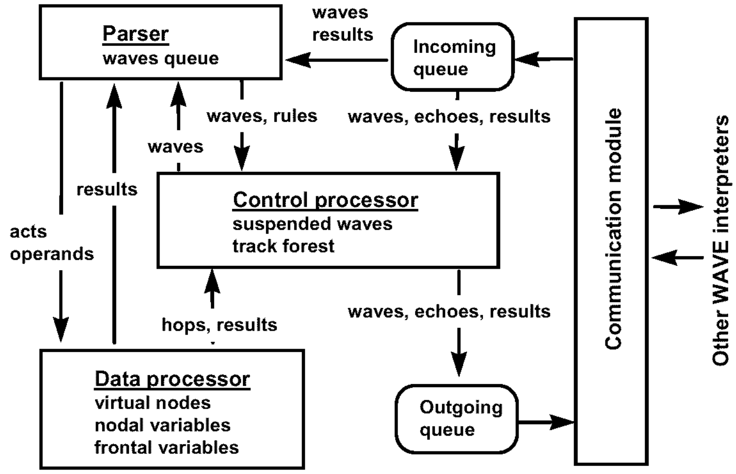

Fig. 5 The WAVE interpreter architecture

4.2 Distributed interpretation in dynamic networks

WAVE interpreters may be stationary, say, as hosts or special coprocessors working via the Internet, or embedded into mobile, both manned and unmanned, platforms. The same space processing formalism allows for a unified coordination of a variety of stationary and mobile systems, as well as their combination.

Predominantly stationary applications of WAVE have been discussed in detail in [1]. Highly dynamic solutions in WAVE may be linked with implementation of flexible parallel scenarios by groups of cooperating mobile robots [2-5], where waves are executed by communicating interpreters which process data related to different nodes in a physical world, and move further if encounter hops to new PW locations. If an interpreter has not completed jobs in some node and has to perform a physical hop, or if a broadcasting hop leads to a number of PW positions, another interpreters may be requested to perform the hops in space, into which the rest of wave may be loaded directly from (and by) the current interpreter (interpreters thus charging each other directly, without an external help). Different strategies of runtime invocation of new interpreters can be used for the evolving spatial scenarios in WAVE [3].

Some general picture of the execution of a unified WAVE scenario in an integration of stationary computer network with mobile manned or unmanned platforms, the latter engaged on a demand of the scenario, may look like the one shown in Figure 6.

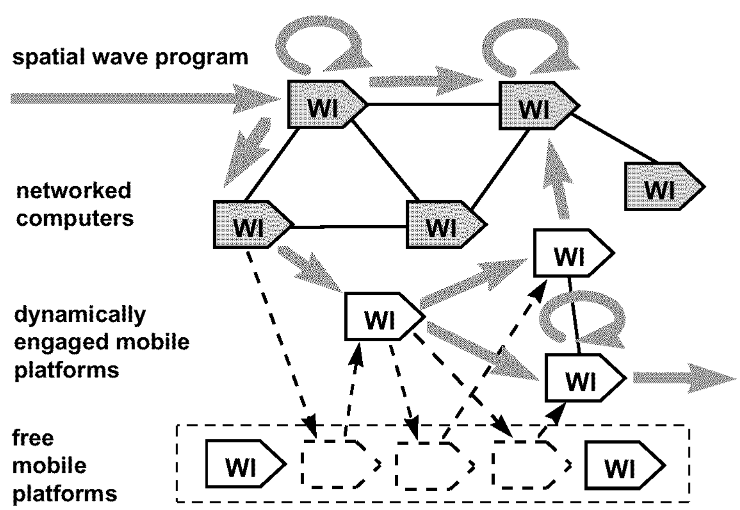

Fig. 6. Unified organization of stationary and mobile networked systems in WAVE

5 Examples of Existing WAVE Applications

Having a long history of development, implementation, and testing in different countries, including world wide experiments via the Internet, WAVE may have efficient applications in a great variety of fields. Some practical examples and results obtained are listed below.

Distributed databases. WAVE was used to create, integrate, and manage large distributed, heterogeneous databases, where any data record can potentially be located in any computer. Integrating dissimilar databases located in different computers and written in other languages, WAVE added an efficient multiuser management layer to the whole system [6,7,9]. Highly parallel and intelligent databases were written purely in WAVE too, allowing for advanced distributed inference and data mining [1,8].

Intelligent network management. Based on a mobile cooperative code freely spreading, self-replicating and recovering in networks, WAVE was efficiently used for management of open computer and communication networks. Having integrated standard network management systems retrieving large amounts of data related to routers and hosts, WAVE was used to extract proper knowledge from the raw data and add top level topology and traffic analysis, improving the overall network performance [10]. A number of key management functions for the cellular networks have been implemented and demonstrated in WAVE, tracking mobile users without (or with minimum use of) central databases [1,11]. Mobile IP protocols, combining the use of computer networks and mobile communications, have been modeled in WAVE too [12].

Distributed interactive simulation. WAVE was used to organize interactive multiuser simulation of large dynamic systems. A distributed system modeling air battles between different types of aircraft was demonstrated, which also integrated aerodynamics modules written in other languages [13-15]. The system comprised dynamic terrain (like radioactive clouds) spreading gradually between computers (and the screens), to be avoided by planes. The system allowed any user to observe both the global battlefield picture and any its local, possibly remote, parts.

Distributed multiuser virtual reality. Creating and processing dynamic distributed knowledge networks, WAVE was used for distributed multiuser virtual reality systems and multicomputer graphics. Efficient integration of the basic VR language, VRML, with WAVE has been implemented and demonstrated [16-18], allowing for dynamic generation, processing, and visualization of VR scenes on many computers. Parallel techniques for creation and transformation of dynamic images in distributed virtual spaces has been programmed and demonstrated [1,19].

Road traffic management. WAVE proved to be useful for advanced road traffic management systems based on distributed computer networks, which are free from traditional bottlenecks caused by the use of centralized or hierarchical databases. A horizontal system describing part of the UK highway network was implemented and demonstrated in WAVE on many computers, with optimal routing for multiple cars, rerouting in case of traffic jams and road damages. The system also simulated the chase of suspected vehicles by police [1].

Modeling collective behavior of robots. Based on mobile cooperative control code spreading in virtual networks, WAVE was successfully used for modeling collective behavior of automatic vehicles moving through space and avoiding obstacles, finding paths through complex mazes, and reaching proper destinations. The vehicles were able to both cooperate and compete for the space and jobs to be done, while pursuing common global goals [1,2].

WAVE public domain. More information about some former projects may be obtained from the WAVE public domain webpages in Germany [20], and UK [21].

6 Organization of Mobile CRF in WAVE

6.1 Advantages of WAVE

Installing communicating copies of the WAVE interpreter in main units of advanced mobile CRF (see Figure 7, in relation to FCS) may provide highest possible integrity and controllability of such systems, which my become capable of performing complex tasks and pursuing both local and global goals in a totally distributed manner, under the guidance of ubiquitous and interactive wave programs.

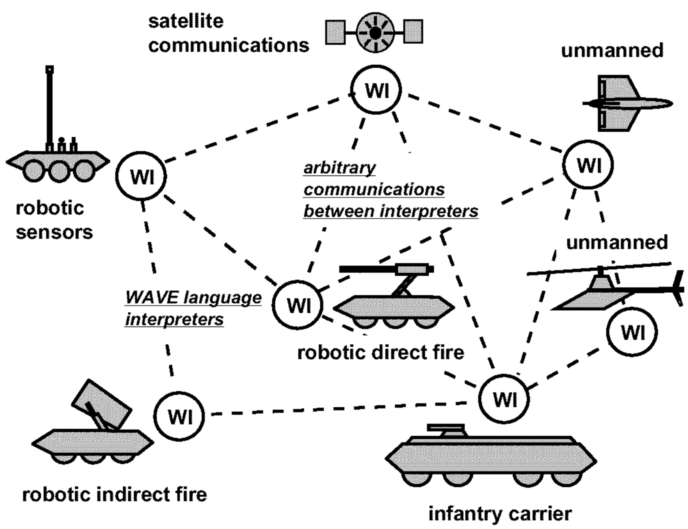

Fig. 7. WAVE interpreter as a universal integration module for advanced mobile Crisis Reaction Forces

WAVE may be particularly useful for solving the following CRF problems:

- Linking distributed heterogeneous military databases.
- Integrating dissimilar command and control systems of different nations into a united C4I infrastructure.
- Self-analysis and self-recovery after indiscriminate failures and damages.
- Support of openness and runtime recomposition and reconfiguration of the international force.
- Automated collection of readiness & operability & statistics, global assessment of distributed situations.
- Efficient integration of unmanned platforms into the force mix.

The following sections provide some elementary examples of using WAVE for the integration of (mobile) CRF.

6.2 Parallel creation and reconfiguration of a united infrastructure

Any topology may be represented in the WAVE syntax in a most compact manner, as a linked graph template, and created in a parallel and fully distributed mode, by deployment and self-evolution & spreading of this template in space. The following program, starting from unit1 (chosen as top of the command hierarchy to be formed), creates oriented links named "infra" between army units, as shown in Fig. 8, in a template flow & unwrapping mode:

```
direct # unit1;
create(
  (+infra ## unit2; +infra ## unit6),
  (+infra ## unit3; +infra ## unit4,
                    +infra ## unit5
  )
)
```

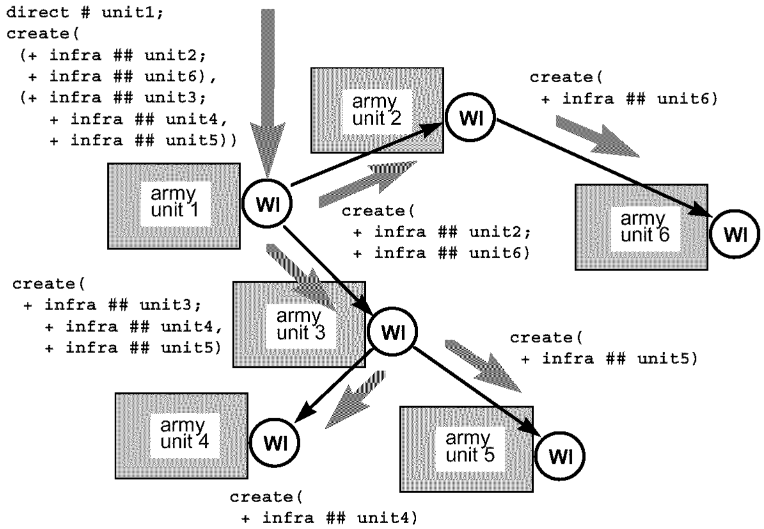

Fig. 8. Runtime creation of a united command and control infrastructure

Any created topology can be easily modified in WAVE by another template which will evolve on the existing topology, dynamically matching it. For example, starting from unit5 and deleting existing link to unit3 in parallel with creation of a new "infra" link to unit2, directed to unit5, the following program-template is sufficient, causing the effect shown in Fig. 9.

```
direct # unit5;
  (any # unit3; LINK = nil),
  create(-infra ## unit2)
```

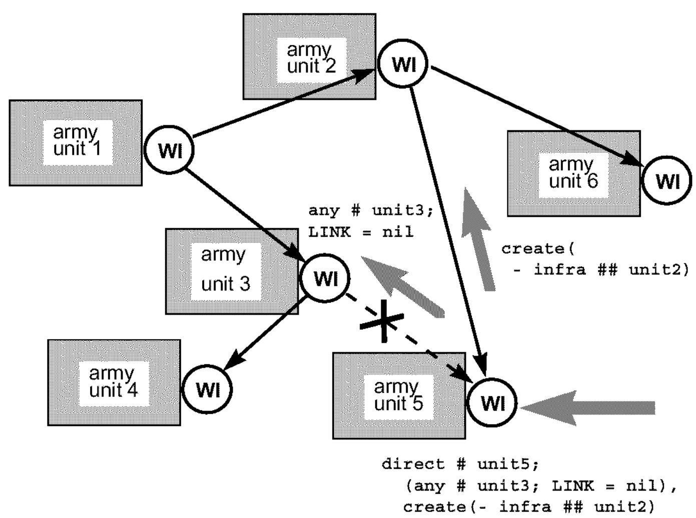

Fig. 9. Runtime reconfiguration of the united infrastructure

6.3 Basic command and control scenario in WAVE

Any command and control systems can be easily created, activated, and simulated in WAVE. The following program, using the above-mentioned infrastructure, implements traditional command and control process, where top level command or mission scenario, applied to the top of hierarchy, is executed in a top-down manner, with acknowledgments moving bottom-up.

At each level, the command is executed locally, according to the peculiarities of this level, and then transformed and modified for the levels below, replicated and sent in parallel to all direct subordinates for further execution & modification. Only after full completion and acknowledgment of the command execution on its and all subordinate levels, a unit reports to its direct superior. The program is based on a recursive navigation procedure `Fcommand_and_control` shown below, which also displays on a terminal confirmation of the acceptance of the command on each level, as well as termination of execution of it on all levels beneath the current level (on different terminals, if the system is distributed).

```
Fcommand_and_control = {
 Flevel += 1;
 sequence(
  TERMINAL = 'entered level: '&& Flevel,
  (Fcommand, Flevel) ? execute_at_level,
  (Fcommand =
    (Fcommand, Flevel) ? transform_detail;
   + infra #; ^ Fcommand_and_control
  ),
  TERMINAL =
    'executed and controlled at and below level: '
   && Flevel
 )
}
```

where external procedures `execute_at_level` and `transform_detail`, taking into account the peculiarity of different command levels, may have fully human execution, human participation & interaction, or be fully automatic. The activation program using this recursive procedure, applied to `unit1` together with the command to be executed by the whole united force, is as follows:

```
Fcommand = <top_command_or_mission_scenario>;
direct # unit1; ^ Fcommand_and_control
```

The distributed hierarchical command and control process, set up by this program, is shown in Fig. 10.

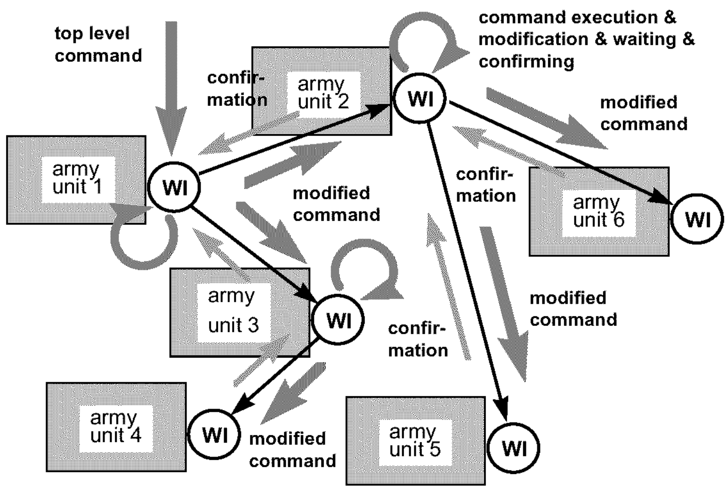

Fig. 10. Traditional hierarchical command and control scenario as a recursive spatial procedure in WAVE

6.4 Solving resource management problems

Let us consider the solution via the created infrastructure of another very vital problem: supply and re-supply of some physical resource to different units of the international force. This resource may be limited, and therefore should be divided proportionally to the extent of local needs, and subsequently physically delivered to proper units from some storage. The following program makes regular top-down checks of the resource demands in units, as a difference between its needed and actual levels, sums up and returns these demands in parallel back through the hierarchy, and analyses on top level the difference between the resource level in the storage and the received sum of demands. Another top-down parallel process makes decisions about the amount of the resource to be supplied to each unit that needs it: If there is enough resource in the storage, the needs are satisfied in full; otherwise the allowed amount for a unit depends on the amount in the central storage, sum of the demands from units, and the unit's demand. After making decision via the control infrastructure, the needed amount of the resource is subsequently physically delivered from the storage directly to the units, to optimize delivery routes (see Fig. 11).

```
Fexplore = {
  + infra #;
  Nrequest = Needed-(Nlocal_resource ? amount);
  Nrequest, ^ Fexplore
};
direct # unit1; Fstart = ADDRESS;
Nglobal_resource = "20 tons of product";
repeat(
 Famount = Nglobal_resource ? amount;
 Frequest_sum = sum(^ Fexplore);
 ( Frequest_sum != 0; Famount != 0;
   or(
    ( Frequest_sum <= Famount;
      repeat(
       + infra #;
       leave(
        Nrequest != 0; Fwithdraw = Nrequest;
        Nlocal_resource +=
          ( direct # Fstart;
            (Nresource, Fwithdraw) ? withdraw
          )
       ), nil
      )
    ),
    repeat(
     + infra #;
     leave(
      Nrequest != 0;
      Fwithdraw=Famount* Nrequest/Frequest_sum;
      Nresource +=
       ( direct # Fstart;
         (Nresource, Fwithdraw) ? withdraw
       )
     ), nil
    )
   ); quit !
 ), 120 ? sleep
)
```

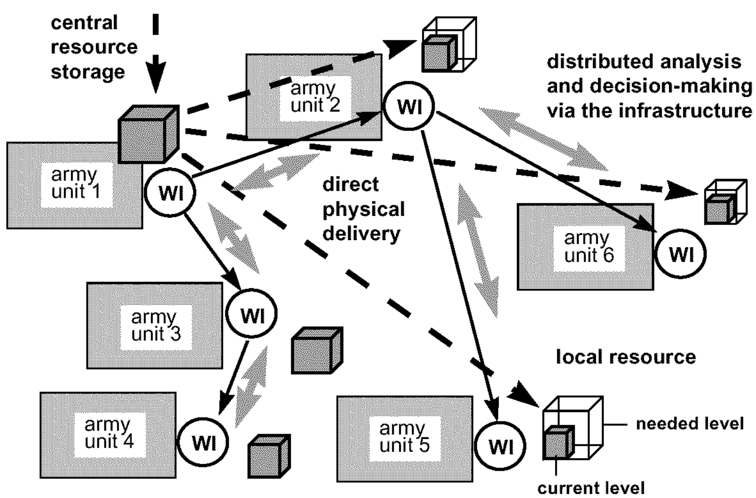

Fig. 11. Resource management via the command infrastructure with direct physical delivery

6.5 A more complex cooperative management example

Cooperative solutions of much more complex problems arising in the united CRF may be effectively organized in WAVE too. An example is shown in Fig. 12, where the initiative starts not from a single point, as in the examples above, but independently from four different points, with four separate (parallel and distributed) optimization scenarios evolving via the infrastructure. Imagine that these scenarios must find a satisfactory solution for all units that launched them (i.e. 1, 4, 5, and 6) by negotiations via the infrastructure, spreading own operations and data to other units if needed. Scenarios 2, 3, and 4 may, for example, reflect local problems in the units that started them, with local vision of their solutions, whereas scenario 1, launched on top level, may moderate solutions for other scenarios, to find an overall optimum. This optimum may have to take into account the results of local optimizations, the latter may need regular re-launching within the global balancing act. All these processes may be highly interactive.

As can be seen from Fig. 12, the locally issued scenarios may invoke a non-local optimization for them as, for example, scenarios 3 and 4 via the superior for them unit2. Scenario 2, starting at unit4, spreads activity one-way only, to unit3, and launches an optimization process for unit4 there, whereas the final solution is found and brought back to unit4 only by the global scenario 1, which regularly navigates the whole hierarchy in both ways, and also coordinates the interaction between scenarios 3 and 4. Efficient program code in WAVE can be easily written for this and many other similar cases.

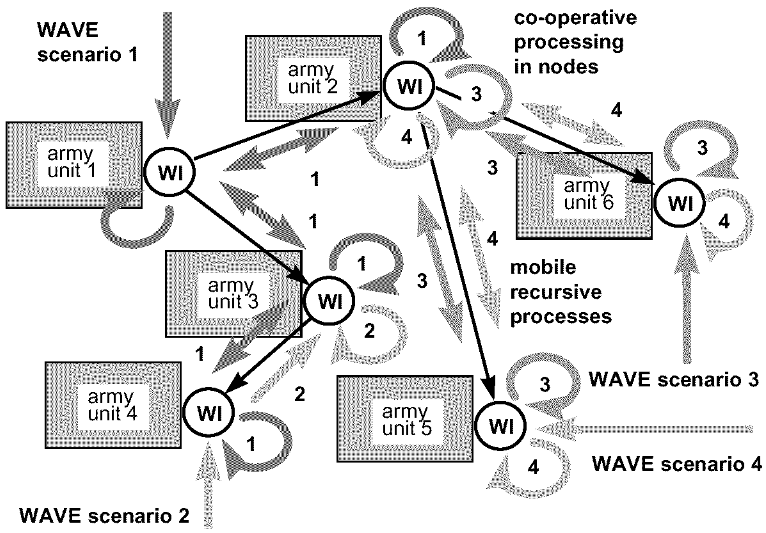

Fig. 12. Cooperative management with multiple interacting scenarios navigating the infrastructure in parallel

7 United Air Defense of CRF in WAVE

7.1 Advantages of using WAVE

WAVE may provide high integration of local air defenses of international forces into a powerful networked system covering the whole region controlled by CRF, with the following possibilities and advantages:

- Simultaneous indissoluble aerial observation of large regions, independent discovery of multiple targets.
- Parallel tracking of many aerial objects and their seamless handover between controlled regions.
- Distributed situation assessment and high collective awareness of the existing aerial threats.
- Intelligent distributed automated, as well as fully automatic, decisions.
- Global optimization of the use of limited international observation and antimissile hardware resources.

- Globally coordinated antimissile fire using networked manned & unmanned ground based and aerial systems.
- Distributed interactive simulation of air defense scenarios, for training troops against the aerial attacks.

Let us consider exemplary solutions of some related problems in WAVE.

7.2 Dynamic forming and updating of a distributed observation infrastructure

Mobile radar stations associated with army units (which may be of short range) may not cover the needed aerial region alone, and may have to communicate frequently to keep the overall observation integral and continuous. To make this communication highly selective and avoid huge network traffic (when each unit communicates with each other one) in tracking aerial objects, very useful may be the establishment of a dynamic neighborhood infrastructure, with virtual links between units reflecting the fact their radar stations cover adjacent (generally overlapping) regions of space, with subsequent communication between the radars only through this infrastructure.

We consider here a program that puts a process into each unit which regularly checks the physical distance from itself to other units, and if it is less than the sum of their radar ranges, a "neighbor" link is set up between the units. On the other hand, if the neighbor link already exists, but the physical distance between nodes exceeds the sum of their radar ranges (i.e. the mobile nodes have moved apart), such a link must be removed. The needed frequency of activation of such a process in each unit, which has to contact all other units in order to maintain a precise enough neighborhood network at each moment, depends on the speed of units, and may not cause serious overhead in the overall system performance, as CRF units are mostly ground-based and their speed is much lower than the speed of aerial objects to be tracked. The following program, working in parallel in all units/nodes, dynamically creates and constantly updates the radar neighborhood infrastructure, where the creation of new links and removal of outdated ones is allowed by only one of the adjacent nodes, to prevent an unnecessary competition:

```
direct # any;
repeat(
  Flocation = WHERE;
  ( direct # any; ADDRESS < PREDECESSOR;
    or(
      ( (Flocation, WHERE) ? distance <= 80.0;
        or( neighbor # PREDECESSOR,
            create(neighbor # PREDECESSOR)
        )
      ),
      (neighbor # PREDECESSOR; LINK = nil); done!
    )
  ),
  300 & sleep
)
```

An example of the created infrastructure is shown in Fig. 13, which may coexist with other virtual infrastructures in WAVE, say, with the command & control one discussed earlier. The created and regularly updated radar neighborhood infrastructure allows us purify, formulate, and solve different discovery, tracking, analysis, handover, decision making, antimissile hardware optimization, and object destruction problems in a distributed network mode, without central resources.

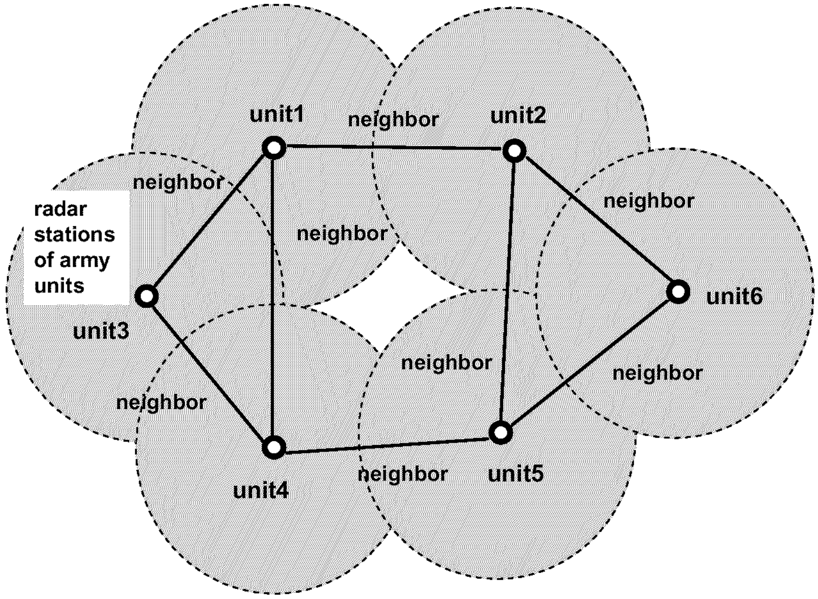

Fig. 13. Runtime creating and updating a radar neighborhood infrastructure

7.3 Simultaneous object tracking and handover between controlled regions

The basic object tracking algorithm using advantages of the mobile wave code, may be as follows. When an intruder object is seen originally by some radar station and identified as a target with distinguishing parameters (type, size, speed, etc.), a mobile tracking process is launched which keeps in view this object and *has the sole authority* of chasing it. Only one such authorized mobile process is created when the target enters the area under CRF control, possibly, in a competition between the stations seeing the target, where closest to the object station may be a winner.

The tracking process regularly checks the vision of the object by the current station, as well as by the neighboring stations by launching subordinate processes in them. If the target gets closer to a neighboring station seeing the target too, the whole process moves itself to this station via the neighborhood infrastructure and continues the target observation there, moving again if the target becomes closer to another neighbor, and so on, thus following the object moving in a physical space via the computer network. If the space coverage by radar stations is not continuos, the chased target may be lost by the tracking process, the latter self-terminating in this case. A new unique tracking process can be launched if the object is rediscovered by some other, non-adjacent, station, which will continue chasing the object via the neighbor links between the radar stations. The following WAVE program, applied initially in all radar stations, implements this algorithm:

```
direct # any;
repeat(
 ( Nold = Nobjects_seen;
   Nobjects_seen = 40 ? observe; Nobjects_seen;
   IDENTITY = VALUE; IDENTITY !~ Nold;
   Fdistance = IDENTITY ? distance;
   or(
    ( neighbor #; IDENTITY ~ Nobjects_seen;
      Fdistance > IDENTITY ? distance
    ),
    release(
     repeat(
      2 ? sleep; Min = infinite;
      sequence(
       ( neighbor #; IDENTITY ~ Nobjects_seen;
         Fdistance = IDENTITY ? distance;
         # PREDECESSOR; Fdistance < Min;
         Min=Fdistance; Mnext=PREDECESSOR; quit!
       ),
       or(
        (IDENTITY ? distance == nil;
         or((Min != infinite; neighbor # Mnext),
            (TERMINAL=IDENTITY && ':lost'; quit!)
         )
        ),
        (Min<IDENTITY?distance; neighbor#Mnext),
        nil
       )
      )
     )
    )
   ); quit !
 ), 1?sleep
)
```

The work of the program is depicted in Fig. 14, with the main mobile tracking process regularly launching subordinate exploration mobile agents checking the neighboring stations. These agents, in their turn, producing echo agents bringing information back to the main process.

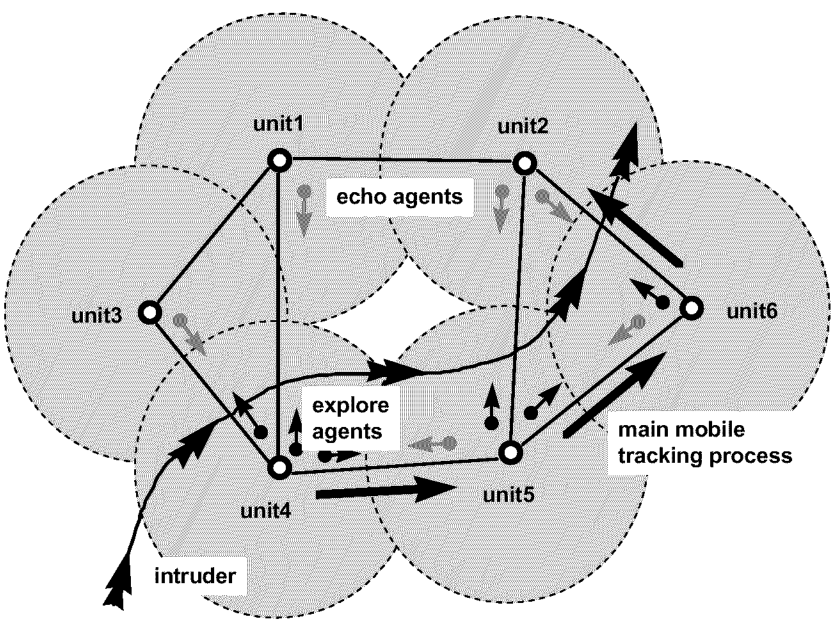

Fig. 14. Distributed networked tracking and handover of aerial objects in WAVE with subordinate mobile agents

The program above describes only the chase in space, and does not specify any payload the tracking process might have, which can easily be added to it. A possible payload (also mobile or being activated as stationary standard procedures in nodes) may include counting the time the object is seen, accumulating its itinerary, measuring its average speed, determining its closeness to sensitive ground-based or aerial components of CRF, etc., in order to decide whether the object is hostile and assess potential threat from it. The final decision to destroy the object can be made, and which hardware is needed for this. New mobile branches of the tracking process may be activated for guiding antimissile rockets via the neighborhood network too, as they may also need crossing boundaries of regions covered by different radar stations, to reach the target. Mobile processes chasing the target and towing the antimissile rockets may cooperatively optimize the collision point, giving a final command for a direct pursuit and destruction. These processes my also check the result, and activate and tow another rockets through the network in case the target survived, and so on. All such dynamic distributed functions can be efficiently implemented in WAVE.

The program discussed above allows for the creation of an unlimited number of mobile tracking processes for different targets, and these processes can develop and migrate simultaneously in the radar station network, as shown in Fig. 15. Any cooperation between individual tracking processes may be provided, also with other, stationary, processes in nodes, to find dynamic optimum solutions for reducing threats and the best use of antimissile weapons. All stationary and mobile processes, including the remote ones, may be highly interactive and may involve human operators in complex situation assessment and decision making processes.

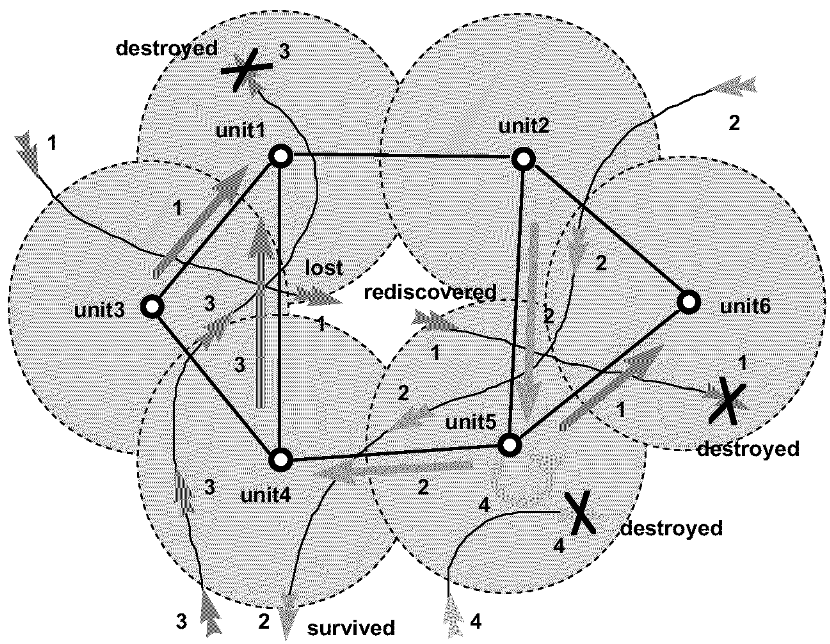

Fig. 15. Simultaneous chasing and destruction of multiple hostile objects in WAVE

A variety of other problems and methods related to integrated distributed air defense of CRF may be solved in WAVE, some of them having already been successfully tested and demonstrated via the Internet. For example, the network of radar stations model was distributed between Germany, UK, and California, and multiple models of alien objects, fighters, and observation, or spying, objects were moving between the continents and countries and interacting with each other throughout the Internet space.

8 Conclusions

As can be seen from this paper, WAVE may serve as a possible basic model and technology for efficient integration and management of advanced multinational CRF and their united air defense systems. It is dynamically deployable, lightweight, mobile, both computational and control networking technology, based on evolving active spatial scenarios or patterns, rather than on communicating agents. Particular hardware or software agents and their interactions emerge dynamically, on the implementation levels only, and only if and when required or available, during the pattern's parallel conquest of space. This allows us to have extremely compact and powerful distributed networked solutions of complex problems (usually about 100 times shorter than in C or Java).

WAVE effectively supports the whole spectrum of system organization levels: from basic network management to the description of high level mission scenarios, distributed interactive simulation, multiuser virtual reality, cooperative robotics, and self-protection from external influences and threats. Due to its volatile, virus-like, fully interpretive nature, WAVE also allows for an efficient self-analysis and self-recovery after indiscriminate damages, as was tested in different projects. Other system models and technologies can be efficiently integrated, expressed and implemented in WAVE, which offers highly integral and seamless solutions for open, dynamic, and heterogeneous systems to which mobile CRF and their air defenses belong.

Most of the existing distributed programming philosophies, and resulting languages and technologies are currently based on the concept of agents [22]. The overall system behavior is considered as a derivative of work of many agents and their multiple interactions, and may often be unpredictable for large dynamic systems, or at least hard to supervise and contain. WAVE, on the opposite, offers a unique opportunity of programming the desired goal-driven whole behavior as a starting point, in the form of high-level active spatial scenario or pattern evolving in space. This may be of paramount importance for advanced military applications which will require highest possible system organization and integrity, in order *to defeat other system organizations, and win the battle*.

Acknowledgments

This paper has been prepared during the author's stay as visiting professor at the Department of Computer Science, University of British Columbia in Vancouver, Canada. Financial support by the Advanced Systems Institute in Vancouver has been appreciated. Encouragement by Prof. A.A. Morozov, Prof. V.P. Klimenko, and Col. A.V. Sotnikov at the National Academy of Sciences of Ukraine was invaluable. Gale Ross, Holly Mitchell, and Sunnie Khuman from UBC helped much in solving different problems in Canada. Also personal thanks to Henry Birkett for useful discussions and write-stimulating home comfort in Vancouver.

References

1. P. S. Sapaty, "Mobile Processing in Distributed and Open Environments", John Wiley & Sons, ISBN: 0471195723, New York, February 1999, 436 p.
2. P. S. Sapaty, "Distributed modeling of cooperative behavior by mobile agents", Proc. Sixth Conference on Computer Generated Forces and Behavioral Representation, IST UCF, Orlando, FL, July 1996.
3. Sapaty, P. S., "Cooperative Conquest of Distributed Worlds in WAVE", Proceedings of the 26th Annual Symposium and Exhibition on Unmanned Systems for the New Millennium AUVSI'99, Baltimore, MD, July 13-15, 1999.
4. Sapaty, P. S., High-Level Spatial Scenarios in WAVE, Proc. of the Fifth Int. Symp. On Artificial Life and Robotics (AROB 5th'00), Oita, Japan, 26-28, January, 2000.
5. Sapaty, P. S., Spatial programming of Distributed Dynamic Worlds, Kluwer Academic Publishers, to be publ. 2000, approx. 300p.
6. Livatharas, C., "Integration of Heterogeneous Databases Using WAVE", MSc Project Report, Dept. EEng, University of Surrey, August 1995.
7. S. Al-Jabir, P. S. Sapaty, M. Underhill, "Integration of heterogeneous databases using WAVE cooperative agents", Proc. First International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agent Technology, The Westminster Central Hall, London, April 1996.
8. Sapaty, P.S., Varbanov, S., Dimitrova, M., "Information systems based on the wave navigation techniques and their implementation on parallel computers", Proc. Intl. Working Conf. Knowledge and Vision Processing Systems, Smolenice, November 1986.

9. Hassanain, E. M., “Some Problems of Management of Distributed Database Systems Using WAVE”, MSc Project Report, Dept. EEng, University of Surrey, Sept. 1995.

10. W. Zorn, P. Sapaty, et al., “Project DAIMON: Global management in open computer networks”, SNI/EK 821 9680/20, Uni. Karlsruhe, 1991.

11. P. S. Sapaty, M. Corbin, P. M. Borst, and A. Went, “WAVE: a new technology for intelligent control in communication networks”, in Proc. Int. Conf. “The Application of RF, Microwave and Millimetre Wave Technologies” (M’94), Wembley, UK, Oct. 25-27, Nexus, 1994.

12. S. Vuong, and L. Mathy, “Simulating the Mobile-IP Protocol Using Wave”, Proc. etaCOM’96, Portland, Oregon, May 1996.

13. P. M. Borst, M. J. Corbin, and P. S. Sapaty, “WAVE processing of networks and distributed simulation”, in Proc. HPDC-3 Int. Conf., San Francisco, Aug. 94, IEEE, 1994.

14. J. C. C. Darling, “Integration of Distributed Simulations of Dynamic Objects”, MSc Research Report, Dept. EEng, University of Surrey, 1995.

15. P. S. Sapaty, P.M. Borst, M.J. Corbin, J.Darling, “Towards the intelligent infrastructures for distributed federations”, Proc. 13th Workshop on Standards for the Interoperability of Distributed Simulations”, IST UCF, Orlando, FL, Sept. 1995.

16. J. C. C. Darling, P. S. Sapaty, “Distributed dynamic virtual reality in WAVE”, Proc. European Simulation Symposium (ESS-96), Genoa, Italy, October 1996.

17. H. K. V. Tan, “Distributed Dynamic 3D Virtual Reality”, MSc Telematics Diploma Project, Dept EEng, University of Surrey, UK, 1997.

18. J. C. C. Darling, “The Application of Distributed and Mobile Computing Techniques to Advanced Simulation and Virtual Reality Systems”, PhD Thesis, Dept. EEng, University of Surrey, September, 1998.

19. P. S. Sapaty, Live demonstration of the WAVE system and applications at the Workshop on Mobile Agents and Security 97, Maryland Center for Telecommunications Research, Department of Computer Science and Electrical Engineering, UMBC, October 27-28, 1997.

20. WAVE in Germany: http://www.ira.uka.de/I32/wave/wave.html.

21. WAVE in UK: http://www.ee.surrey.ac.uk/Research/DKP/.

22. Minsky, M. L., “The society of mind”, Simon & Schuster, 1988.